In re Patent Application of
CALABRO' ET AL.
Serial No. **Not Yet Assigned**
Filed: **Herewith**
________________________________/

In the Claims:

Claims 1-5 (Cancelled).

6. (New) A method for performing a Shor's quantum algorithm as a function ($f(x)$) encoded with n qubits, the method comprising:

performing a superposition operation according to the Shor's quantum algorithm over a set of input vectors, and generating a corresponding superposition vector, the performing comprising

calculating as a function of the n qubits a value ($1/2^{n/2}$) of non-null components of the superposition vector, and

calculating indices (i) of the 2^n non-null components of the superposition vector as an arithmetic succession, a seed of which is 1 and a difference of which is 2^n ($i=1+2^n(j-1)$);

performing an entanglement operation on the superposition vector, and generating a corresponding entanglement vector; and

performing an interference operation on the entanglement vector, and generating a corresponding output vector.

7. (New) A method according to Claim 6, wherein performing the entanglement operation comprises:

calculating indices (k) of the 2^n non-null components of the entanglement vector, summing to each term of the arithmetic succession a relative number corresponding to the value of the function ($f(j)$) calculated based upon a position

(j) of the term in the the succession ($k=f(j)+1+2^n(j-1)$); and

a value of the non-null components of the entanglement vector being equal to the non-null components of the superposition vector.

8. (New) A method according to Claim 7, further comprising generating real and imaginary components of the output vector by performing the following:

for each index *h* of the real and imaginary components, veryfying whether among terms of the arithmetic succession $h \bmod 2^n+1+2^n(j-1)$ has a seed of $h \bmod 2^n+1$, with *j* being an index and 2^n being a common difference, that there is at least one term corresponding to an index of the non-null component of the entanglement vector; and if the verifying is negative, making the real and imaginary components equal to zero;

otherwise calculating the real component as a product between a value of the non-null components and a summation of the following cosine functions

$$\cos\left(2\pi\frac{(j-1)\cdot \mathrm{int}[(h-1)/2^n]}{2^n}\right),$$ and

calculating the imaginary component as a product between a value of the non-null components and a summation of the following sine functions

$$\sin\left(2\pi\frac{(j-1)\cdot \mathrm{int}[(h-1)/2^n]}{2^n}\right)$$

for all values of the index j of the arithmetic succession in which indices (k) of the non-null components of the entanglement vector correspond thereto.

9. (New) A method for performing a Simon's quantum algorithm as a function ($f(x)$) encoded with n qubits, the method comprising:

performing a superposition operation according to the Simon's quantum algorithm over a set of input vectors, and generating a corresponding superposition vector, the performing comprising

calculating as a function of the n qubits a value ($1/2^{n/2}$) of non-null components of the superposition vector, and

calculating indices (i) of the 2^n non-null components of the superposition vector as an arithmetic succession, a seed of which is 1 and a difference of which is 2^n ($i=1+2^n(j-1)$);

performing an entanglement operation on the superposition vector, and generating a corresponding entanglement vector;

performing an interference operation on the entanglement vector, and generating a corresponding output vector.

10. (New) A method according to Claim 9, wherein performing the entanglement operation comprises:

calculating indices (k) of the 2^n non-null components of the entanglement vector, summing to each term of the

arithmetic succession a relative number corresponding to the value of the function ($f(j)$) calculated based upon a position (j) of the term in the the succession ($k=f(j)+1+2^n(j-1)$); and

a value of the non-null components of the entanglement vector being equal to the non-null components of the superposition vector.

11. (New) A quantum gate for performing a Shor's quantum algorithm as a function ($f(x)$) encoded with n qubits, the quantum gate comprising:

a superposition subsystem for performing a superposition operation according to the Shor's quantum algorithm over a set of input vectors, and generating a corresponding superposition vector, said superposition subsystem

calculating as a function of the n qubits a value ($1/2^{n/2}$) of non-null components of the superposition vector (P), and

calculating indices (i) of the 2^n non-null components of the superposition vector as an arithmetic succession, a seed of which is 1 and a difference of which is 2^n ($i=1+2^n(j-1)$);

an entanglement subsystem for performing an entanglement operation on the superposition vector, and generating a corresponding entanglement vector; and

an interference subsystem for performing an interference operation on the entanglement vector, and generating a corresponding output vector.

12. (New) A quantum gate according to Claim 11,

further comprising a first memory buffer for storing the value ($1/2^{n/2}$) and the indices (i).

13. (New) A quantum gate according to Claim 11, wherein said entanglement subsystem calculates indices (k) of the 2^n non-null components of the entanglement vector, sums to each term of an arithmetic succession a number corresponding to a value of the given function ($f(j)$) calculated based upon a position (j) of the term in the succession ($k=f(j)+1+2^n(j-1)$); and a value of the non-null components of the entanglement vector being equal to the non-null components of the superposition vector.

14. (New) A quantum gate according to Claim 13, further comprising a second memory buffer for storing the indices (k) of the 2^n non-null components of the entanglement vector.

15. (New) A quantum gate according to Claim 13, wherein said interference subsystem generates real and imaginary components of the output vector by performing the following:

for each index h of the real and imaginary components, veryfying whether among terms of the arithmetic succession $h \bmod 2^n + 1 + 2^n(j-1)$ having a seed of $h \bmod 2^n + 1$, with j being an index and 2^n being a common difference, that there is at least one term corresponding to an index of the non-null component of the entanglement vector; and if the verifying is negative, making the real and imaginary components equal to zero;

otherwise calculating the real component as a product between a value of the non-null components and a summation of the following cosine functions

$$\cos\left(2\pi\frac{(j-1)\cdot\operatorname{int}[(h-1)/2^n]}{2^n}\right), \text{ and}$$

calculating the imaginary component as a product between a value of the non-null components and a summation of the following sine functions

$$\sin\left(2\pi\frac{(j-1)\cdot\operatorname{int}[(h-1)/2^n]}{2^n}\right)$$

for all values of the index j of the arithmetic succession in which indices (k) of the non-null components of the entanglement vector correspond thereto.

16. (New) A quantum gate for performing a Simon's quantum algorithm as a function $(f(x))$ encoded with n qubits, the quantum gata comprising:

a superposition subsystem for performing a superposition operation according to one of the quantum algorithms over a set of input vectors, and generating a corresponding superposition vector, said superposition subsystem

calculating as a function of the n qubits a value $(1/2^{n/2})$ of non-null components of the superposition vector, and

calculating indices (i) of the 2^n non-null

components of the superposition vector as an arithmetic succession, a seed of which is 1 and a difference of which is 2^n ($i=1+2^n(j-1)$);

an entanglement subsystem for performing an entanglement operation on the superposition vector, and generating a corresponding entanglement vector; and

an interference subsystem for performing an interference operation on the entanglement vector, and generating a corresponding output vector.

17. (New) A quantum gate according to Claim 16, further comprising a first memory buffer for storing the value ($1/2^{n/2}$) and the indices (i).

18. (New) A quantum gate according to Claim 16, wherein said entanglement subsystem calculates indices (k) of the 2^n non-null components of the entanglement vector, sums to each term of an arithmetic succession a number corresponding to a value of the given function ($f(j)$) calculated based upon a position (j) of the term in the succession ($k=f(j)+1+2^n(j-1)$); and a value of the non-null components of the entanglement vector being equal to the non-null components of the superposition vector.

19. (New) A quantum gate according to Claim 18, further comprising a second memory buffer for storing the indices (k) of the 2^n non-null components of the entanglement vector.